SANDXXXX–XXXX
Unlimited Release
Printed December XXXX

Acro User Manual
Version 1.0

William E. Hart
Discrete Algorithms and Mathematics Department
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM
http://www.cs.sandia.gov/~wehart
wehart@sandia.gov

Abstract This document provides a user's guide for the Acro software library. The Acro Project is an effort to facilitate the design, development, integration and support of optimization software libraries. The goal of the Acro Project is to develop (parallel) optimization solvers and libraries using object-oriented software frameworks that facilitate the application of these solvers to large-scale engineering and scientific applications. Thus Acro includes both individual optimization solvers as well as optimization frameworks that provide abstract interfaces for flexible interoperability of solver components. Acro has been written to support optimizers that are broadly useful for DOE's engineering and scientific applications. In particular, many solvers included in Acro can exploit parallel computing resources to solve optimization problems more quickly.

acro - From Greek akros, at the point, end, or top.

acro - A generic term for warblers of the genus *Acrocephalus*, usually refering to the sedge and/or reed warblers. The sedge warbler is a small, quite plump, warbler with a striking broad creamy stripe above its eye, and greyish brown legs. It is brown above with blackish streaks and creamy white underneath. It spends summers in the UK and winters in Africa, south of the Sahara Desert. Its song is a noisy, rambling warble compared to the more rhythmic song of the reed warbler.

acro - A Common Repository for Optimizers

1 Introduction

This document provides a user's guide for the Acro software library. The Acro Project is an effort to facilitate the design, development, integration and support of optimization software libraries. The goal of the Acro Project is to develop (parallel) optimization solvers and libraries using object-oriented software frameworks that facilitate the application of these solvers to large-scale engineering and scientific applications. Thus Acro includes both individual optimization solvers as well as optimization frameworks that provide abstract interfaces for flexible interoperability of solver components. The optimization packages in Acro have been written to support optimizers that are broadly useful for DOE's engineering and scientific applications. In particular, many solvers included in Acro can exploit parallel computing resources to solve optimization problems more quickly.

The Acro framework uses a two level software structure that connects a system of *packages*. An Acro package is usually developed by a (relatively) small set of developers to solve a specific set of problems. Packages exist beneath the Acro top level, which provides a common look-and-feel. Each package has its own structure, documentation and set of examples, and it is possibly available independently of Acro. However, Acro provides a framework for exploiting synergy between packages in a flexible manner.

Acro currently includes about ten different packages, several of which include interfaces to external software libraries. The entire set of packages covers a variety of algorithmic and numerical methods for optimization, as well as utilities that facilitate the development of software for scientific computing. A detailed description of each Acro package is beyond the scope of this document. Instead, we provide a birds-eye view of the Acro packages and their corresponding solvers. We also include information about how Acro can be installed and used. Further, we provide pointers to additional package documentation.

2 Software Overview

Acro integrates a variety of optimization software packages, including libraries developed at Sandia National Laboratories as well as publicly available third-party libraries. The goal of this integration is to provide a single framework that includes a wide range of optimization methods.

2.1 Capabilities

Three classes of optimizers are well-represented in Acro:

1. branch-and-bound methods for integer programming
2. heuristic global optimization methods
3. derivative free local optimization methods

2.1.0.1 Integer Programming The **PICO** package defines an object-oriented scalable library for branch-and-bound. In particular, **PICO** includes a well-developed parallel mixed integer linear programming (MILP) solver. **PICO** depends on **UTILIB** for many basic data structures and

system utilities. Additionally, **PICO** uses the **COIN** interface library to encapsulate linear programming solvers. Two publicly available linear programming solvers are integrated into Acro: **SOPLEX** and **CLP** (which is included in **COIN)**. The **mpPCx** parallel linear programming solver is also being developed at Sandia (this is currently not part of Acro).

PICO has been integrated with the **GLPK** package to support a public domain interface to AMPL integer programming problems. **PICO** has also been integrated with the **AMPL** package, which includes the AMPL routines for setting up an AMPL solver.

2.1.0.2 Global Optimization The **COLIN** package defines a general-purpose interface for optimization solvers. Although many C++ optimization software libraries have been developed, these designs invariably incorporate specific features that facilitate the development and application of the author's algorithms. By contrast, COLIN is intended to provide a generic interface for optimizers that is simple to use and which can encapsulate the basic algorithmic interface of a wide range of general-purpose optimizers. As such, COLIN can be used as 'middle-ware' to facilitate the interface of an optimizer to (a) other optimizers to form hybrid solution strategies and (b) to generic user-interfaces like AMPL or Excel, for which well-defined APIs exist.

The **Coliny** package defines a variety optimizers using COLIN. In particular, **Coliny** includes global optimization heuristics like multi-start local search, evolutionary algorithms and DIRECT. Additionally, **Coliny** includes COLIN interfaces to several other global optimization solvers, including a Lipshitzian solver defined using **PICO**. The **3po** package includes COLIN interfaces for a variety of other publicly available optimizers.

2.1.0.3 Derivative Free Optimization Derivative free optimizers perform local optimization on continuous search domains without the aid of gradients or other derivative information. The **Coliny** solver includes several derivative free methods, and in particular a variety of pattern search methods. The **APPSPACK** package defines a parallel pattern search method that exploits a novel asynchronous parallel search.

2.2 End-User Applications

The principal goal of Acro is to provide support of optimization libraries. However, there are several end-user applications that can be build within Acro.

1. **PICO** - a solver for mixed-integer linear programming problems. **PICO** can be use with AMPL, or with user-input files in standard formats (LP, MPS, etc).
2. **coliny** - a solver for derivative-free and heuristic global optimization. **coliny** is principally indended for use with AMPL.

Documentation for the use of these solvers is currently being developed.

2.3 Overview of Acro Packages

Acro is organized around a set of closely related software packages. Figure 1 illustrates the dependencies between these packages; the dashed lines indicate packages and dependencies that are currently being developed.

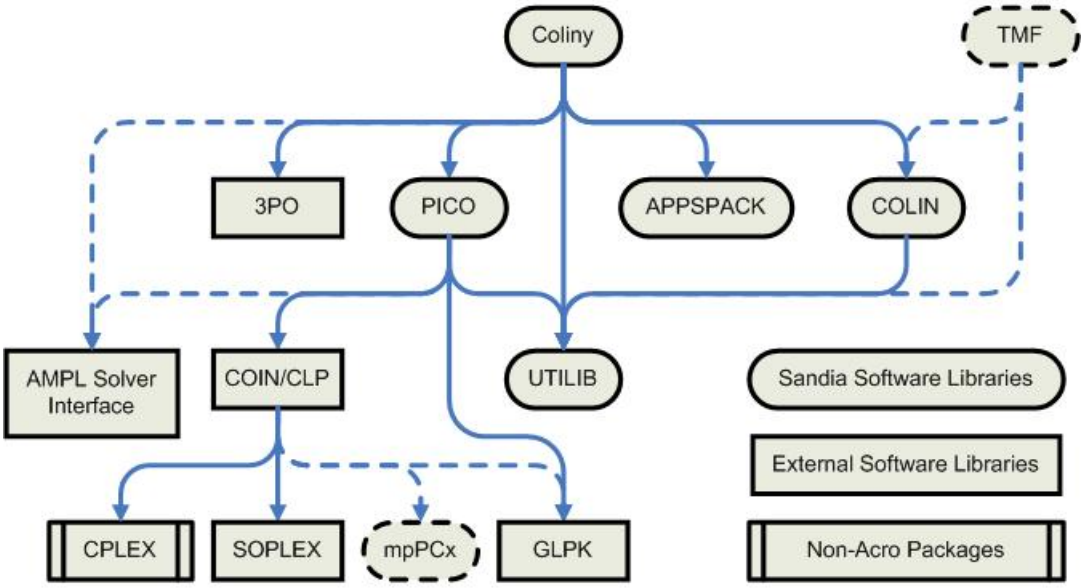

Figure 1: An illustration of inter-package dependencies within Acro.

1. **3po** - A library of third party optimizers that are publicly available.

2. **CMO** - An example of the use of the MW software for parallel master-worker. This is not a core Acro package, and thus it will be reconstituted shortly.

3. **COIN** - A subset of the COIN optimization repository that includes linear and integer programming interfaces for a variety of third-party software projects. This package uses the Acro configure/makefile's, since COIN does not have a sufficiently flexible configure/build process. See `http://www.coin-or.org` for further information on this software.

4. **SOPLEX** - The SOPLEX linear programming library. SOPLEX contains an implementation of the revised simplex algorithm. It features primal and dual solving routines for linear programs and is implemented as a C++ class library that can be used with other programs. See `http://www.zib.de/Optimization/Software/Soplex/` for further information on this software.

5. **ampl** - This package includes C files provided by AMPL to setup solvers that can be used by AMPL. In particular, this package provides the ability to read the AMPL $*$.nl file format.

6. **appspack** - An asyncronous parallel pattern search method developed at Sandia National Laboratories. See `http://software.sandia.gov/appspack/` for further information on this software.

7. **colin** - A general-purpose C++ optimization interface developed at Sandia National Laboratories. This library provides a middle-ware layer for optimizers that facilitates the use of hybrid optimizers, as well as optimizers applied to novel, user-defined search domains.

8. **coliny** - Coliny is a collection of C++ optimizers that are derived directly from the COLIN interface. This library includes methods such as pattern search, evolutionary algorithms,

multistart local search and DIRECT. Coliny is a revision of the older SGOPT library, which is now deprecated.

9. **glpk** - The GNU linear programming toolkit. GLPK is intended for solving large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. See `http://www.gnu.org/software/glpk/glpk.html` for further information on this software.

10. **pico** - PICO defines a flexible branch-and-bound engine that uses an object-oriented model to allow it to be easily adapted to new applications. A novel feature of PICO is its ability to scale up parallel branch-and-bound to thousands of processors. PICO also includes solvers for mixed-integer linear programming.

11. **sgopt** - SGOPT defines a collection of C++ optimizers. This software has been superceded by Coliny, and its use is now deprecated.

12. **tmf** - TMF is a Templatized Metaheuristics Framework that is under development.

13. **utilib** - UTILIB is a library of C and C++ utility methods. It includes classes of standard data structures, general-purpose mathematical routines, sorting methods, random number generates, tools for processing command-line options and tools to support parallelization of software.

3 Software Management

3.1 Downloading

The Acro software can be downloaded as a compressed tarfile from `http://software.sandia.gov/Acro/`.

Alternatively, Acro can be checked out from the Acro CVS (Concurrent Version System) repository. The CVS respository for Acro can be accessed by executing

```
cvs -d :ext:${userid}@software.sandia.gov:/space/CVS-Acro checkout acro
```

This requires the use of an account (userid) on **software.sandia.gov**. Accounts on this machine are generally restricted to Sandians and academic collaborators working with Sandians. The scripts **cvs.a** and **ssh.cvs** can be used to encapsulate the access to this repository in a convenient manner; these scripts can be downloaded from `ftp://ftp.cs.sandia.gov/pub/papers/wehart/src/shells`.

By default, the **cvs.a** script uses the unix command **whoami** to determine the user id for the cvs connection. In general, the id ${userid} on software.sandia.gov will be selected by Sandia administrators. If this does not match your id provided by **whoami**, then you can set the **ACRO_USER** environmental variable to be id on software.sandia.gov. For example, within a csh environment, you would do

```
setenv ACRO_USER janedoe
cvs.a checkout acro
```

to checkout Acro from the account `janedoe@software.sandia.gov.`

The Acro CVS repository is organized into different modules that include different subsets of the Acro packages:

- **acro** - The Acro repository with all software packages and an integerated configuration facility that builds them
- **acro-coliny** - The Acro repository with the Coliny software package and all packages that it depends on
- **acro-pico** - The Acro repository with the PICO software package and all packages that it depends on

3.2 Online Resources

The CVS repository for Acro is on software.sandia.gov. This machine supports a variety of tools to facilitate the development of software tools, including

1. Bugzilla (`http://software.sandia.gov/bugzilla/`) - online bug reporting
2. Mailman (`http://software.sandia.gov/mailman/listinfo/`) - mailing list management
3. Bonsai (`http://software.sandia.gov/bonsai/cvsqueryform.cgi`) - a query interface to CVS

The following mailing lists are supported for Acro:

1. `acro-announce@software.sandia.gov` - general announcements
2. `acro-bugs@software.sandia.gov` - bug reports
3. `acro-checkins@software.sandia.gov` - CVS checking reports
4. `acro-developers@software.sandia.gov` - a developers mailing list
5. `acro-help@software.sandia.gov` - a help list
6. `acro-misc-checkins@software.sandia.gov` - CVS checking reports for miscellaneous acro components
7. `acro-users@software.sandia.gov` - a general users' mailing list

The mailman server also includes some mailing lists for specific Acro components like APPSPACK, PICO and Coliny.

4 Installation

4.1 Standard UNIX Installation

After the Acro software has been downloaded, it must be configured for building on specific hosts for specific target platforms. Installation on UNIX systems is performed by the following steps:

1. If you have an archive, unpack it as follows

```
gunzip acro-$VERSION.tar.gz    # uncompress the archive
tar xf acro-$VERSION.tar       # unpack it
```

2. Move into the **acro** directory and run the **configure** script.

```
cd acro
./configure
make
```

   This will configure the software for your machine and compile the Acro library (see Section **Configuration** (p. 12) for more details concerning configuration options and the details of the configuration process). After building Acro, the Makefile scripts will create links to package header directories in the directory **acro/include**, and links to all package libraries are included in **acro/lib**. Note that the makefiles in Acro may not be portable to all **make** commands. However, they do work with the GNU **gmake** command.

You can remove object files, libraries, and executables by typing

```
make clean
```

If you wish to re-configure your Acro source from scratch or re-generate all custom makefiles, type

```
make distclean
```

This will remove all symbolic links, custom makefiles, and **config.status** files. This will require a re-configure of the system and is usually done in preparation for updating source in the CVS repository or creating tape archives for distribution.

4.2 Custom UNIX Installation

Acro can also be built with the following custom configure/build process

```
cd acro
./setup configure build
```

This custom configure/build process overcomes a common problem with the standard configure/build process: it can create a voluminous output that is difficult to follow and which can mask the fact that there errors have occured. This is a particular issue for large projects like Acro, which has many independent packages that may fail to configure or build.

The **setup** command performs the following actions

1. Pipe the configuration output into acro/test/config.out and generate a test/config.xml summary file
2. Pipe the build output into acro/test/build.out and generate a test/build.xml summary file (this includes builds of Acro executables and test executables)

The XML files generated in the acro/test director provide a quick reference to confirm that the software was configured and built properly. The use of **setup** can be extended to include configuration options, and to perform tests. For example the following configures using MPI and performs fast smoke tests of various Acro packages:

```
cd acro
./setup configure --with-mpi  build  smoke
```

4.3 Setup Documentation

The main Acro makefile includes several targets for generating documentation. This documentation is generated with the **doxygen** utility, so documentation can be generated in several formats. The command

```
make html
```

creates HTML documentation, and the command

```
make pdf
```

creates documentation in a single PDF file. This later command requires the use of **latex**, **dvips** and **ghostscript** packages.

5 Acro Licenses

The Acro framework is licensed with the GNU lesser general public license (see below). Acro includes a variety of third-party software packages, which have separate licensing policies. These licenses are included here to provide a single point of reference for the use of Acro. The following list describes the licenses for the different Acro packages:

1. UTILIB, PICO, SGOPT, Coliny, COLIN, APPSPACK, glpk

   The GNU Lesser General Public License (p. 24)

   academic, commercial
2. Any.h (utilib/src/adt)

   BOOST License (p. 16)

   academic, commercial

3. RANLIB (utilib/src/ranlib)

   RANLIB License (p. 16)

   academic, commercial

4. COIN and CLP

   Common Public License (p. 17)

   academic, commercial

5. SOPLEX

   ZIB Academice License (p. 21)

   academic

6. ampl

   Lucent Technologies License (p. 23)

   academic, commercial

These licenses are provided in Appendix **Licensing Details** (p. 16) generally allow for unrestricted academic use. The SOPLEX license is restricted to academic users, but the CLP linear programming license is available for commercial use.

Please contact the original software developers if you have questions concerning the use of these software components. Further, you can contact `acro-help@software.sandia.gov` for assisstance if you decide that you wish to build Acro without some particular software component.

6 References

The following are sources of package-specific documentation that provide more detail on the use of Acro.

1. W. E. Hart. UTILIB user manual version 1.0. Technical Report SAND2001-3788, Sandia National Laboratories, 2001.

7 Acknowledgements

The Acro Project grew out of the ongoing development of the UTILIB, SGOPT, Coliny, and PICO libraries within the Discrete Algorithms and Mathematics Department. In particular, the development of PICO and Coliny increasingly required the integration of third-party libraries. Although a variety of people have contributed to the Acro packages, I would like to thank Cindy Phillips, Jonathan Eckstein, Mario Alleva, Shannon Brown and Mike Eldred for their input on these software libraries, which ultimately influenced the design of Acro itself. The design of Acro has been significantly influenced by their feedback.

This work was supported in part by the Mathematics, Information and Computational Science program, U.S. Department of Energy, Office of Energy Research, as well as the Laboratory

Directed Research Program at Sandia National Laboratories. This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

This document was prepared using the Doxygen software documentation tool, developed by Dimitri van Heesch, copyright 1997-2004.

We are grateful to Mike Langman for graciously allowing us to use his beautiful Sedge Warbler illustration. Other illustrations and photographs by Mike Langman are available at `Mike Langman Bird Art`.

A Configuration

A.1 Overview

The **configure** script automates much of the setup activity associated with building large suites of programs like Acro on various hardware platforms. This includes

1. making symbolic links so that files used for configuration can be accessed from one location
2. generating Makefiles so that objects, libraries, executables and other 'targets' can be created for specific and unique hardware platforms
3. calling itself recursively so that sub-directories can also be configured

There are a number of configuration options that can be used to customize the installation. The core parameter list for the **configure** script is:

```
configure hosttype [--target=target] [--srcdir=dir] [--rm]
                   [--site=site] [--prefix=dir] [--exec-prefix=dir]
                   [--program-prefix=string] [--tmpdir=dir]
                   [--with-package[=yes/no]] [--without-package]
                   [--enable-feature[=yes/no]] [--disable-feature]
                   [--norecursion] [--nfp] [-s] [-v] [-V | --version]
                   [--help]
```

Many of these options are not necessary since system information can be often acquired from your local machine. Refer to the Cygnus **configure** documentation for complete information.

A.2 Standard Options

The following options are either commonly used or specific to Acro packages:

[–with-compiler=<gcc,CC>]	Sets up a specific compiler; The native compiler is the default.
[–target=<solaris>]	Optional flag to specify the target machine that you are cross-compiling for.
[–site=<snl980>]	Specifies the site-specific locations for MPI, etc.
[–with-debugging]	Turns on the DEBUGGING macro and sets the OPTIMIZATION macro to <flag> (code is compiled with -g by default).
[–with-mpi]	Turns on the use of the MPI package.
[–with-mpe]	Turns on the use of the MPE package.
[–with-swig]	Enables the use of swig to wrap Acro packages for use with the Python scripting language.
[–with-static]	Enables the compilation of statically linked libraries (the default).
[–with-insure]	Enables the compilation with the insure++ debugging tool.
[–with-shared]	Enables the compilation of dynamically linked libraries, which can be shared.
[–with-optimization=<level>]	Sets the optimization level used when compiling the source files. This is overridden by the –with-debugging flag.
[–with-gprof]	Sets the compiler to use flags that force executables to dump output files that can be read by the unix **gprof** utility to profile the code.

A.3 Details of the Configuration Process

The **configure** script creates Makefiles from **Makefile.in** template files, which outline the basic 'targets' that need to get built. This script can be customized using the **configure.in** file. Variables that are package, site or hardware dependent are stored in individual 'fragment' files. These 'fragment' files are added to the custom created Makefiles using the host, target, package and/or site parameters. After the **configure** command is completed, three files will be generated in each configured directory.

1. Makefile-${target}

   The suffix, ${target}, will depend on the target architecture, which defaults to the host architecture; native builds have identical host and target values.

2. Makefile

   This will be a symbolic link to **Makefile-${target}**. Consequently, when **make** is executed, the default behavior is to use the last generated Makefile-${target} file.

3. config.status

   A 'recording' of the configuration process (i.e., what configure options were used to generate the current makefile).

Running **configure** takes a while, so be patient. Verbose output will always be displayed unless the parameter '–silent' is used. Further, the parameter '–norecursion' can be used to configure only the current directory. All generated **config.status** files include this parameter as a default for easy makefile re-generation; after editing a **Makefile.in** file, you can construct the associate Makefile file by executing **config.status**.

The **configure** command supports multi-platform environments using Makefile fragment files. The fragment files for these platforms and for the packages that Acro relies on are located in the **acro/config** directory. There are five types of files in this directory:

```
mh-<host>
Fragments that define the utilities provided by the host (e.g. the
definition of MAKE).

mp-<target>-<site>
Fragments that define information for the packages that are used by
Acro (e.g. MPI).

ms-<site>
Fragments that define the site-specific general configuration
information. If this does not exist for a given site, then the
default ms-default fragment is used.

mt-<target>
Fragments needed to specfy how to compile code for a target
architecture (e.g. compiler name/location).

mf-<host>-<target>-<site>
Automatically generated by the configure scripts.
```

A.4 Platforms Supported

Acro can be configured for code development and execution on the following platforms:

SUN	Solaris 2.8	CC v5.4	solaris
PC	Red Hat LINUX 9	g++ v3.2.2	linux
SGI	IRIX 6.5	CC v7.41	irix
DEC	OSF 5.1	cxx v6.3-002	osf
IBM	AIX 5.1	xlC v6	aix
TFLOP	Cougar (compute)	ciCC v4.4.0	cougar

```
TFLOP       OS (service)            iCC  v4.4.0     tflop
Cplant      Compute OS              c++             cplant
Cplant      Service OS              g++             cplant
PC          Windows (Cygwin)        g++             cygwin
Mac         OSX (Darwin)            g++             darwin
HP          HP-UX 10.20             CC              hpux
```

The last column shows the configuration target used to configure Makefiles for each of these systems.

The standard configure and build process described in Section **Installation** (p. 8) works on almost all of these platforms. A noteable exception are cougar builds, which generate executables for the ASCI Red compute nodes. These builds need to performed with a cross-compiler. For example, when building on sasn100.sandia.gov, you would configure as follows:

```
./configure --site=sasn100 --target=cougar
```

B Licensing Details

Acro includes a variety of third-party software packages, which have separate licensing policies. These licenses are included here to provide a single point of reference for the use of Acro. These licenses generally allow for unrestricted academic use. The SOPLEX license is restricted to academic users, but the CLP linear programming license is available for commercial use.

B.1 BOOST License

```
// Copyright Kevlin Henney, 2000, 2001, 2002. All rights reserved.
//
// Permission to use, copy, modify, and distribute this software for any
// purpose is hereby granted without fee, provided that this copyright and
// permissions notice appear in all copies and derivatives.
//
// This software is provided "as is" without express or implied warranty.
```

B.2 RANLIB License

UTILIB includes elements of the **ranlib** library, which defines a variety of random number generators. The following licensing information comes with **ranlib**:

```
                              LEGALITIES

Code that appeared  in an    ACM  publication  is subject  to    their
algorithms policy:

     Submittal of  an  algorithm    for publication  in   one of   the  ACM
     Transactions implies that unrestricted use  of the algorithm within  a
     computer is permissible.   General permission  to copy and  distribute
     the algorithm without fee is granted provided that the copies  are not
     made  or   distributed for  direct   commercial  advantage.    The ACM
     copyright notice and the title of the publication and its date appear,
     and  notice is given that copying  is by permission of the Association
     for Computing Machinery.  To copy otherwise, or to republish, requires
     a fee and/or specific permission.

     Krogh, F.  Algorithms  Policy.  ACM  Tran.   Math.  Softw.   13(1987),
     183-186.

We place the Ranlib code that we have written in the public domain.

                                NO WARRANTY

     WE PROVIDE ABSOLUTELY  NO WARRANTY  OF ANY  KIND  EITHER  EXPRESSED OR
     IMPLIED,  INCLUDING BUT   NOT LIMITED TO,  THE  IMPLIED  WARRANTIES OF
     MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK
     AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS  WITH YOU.  SHOULD
     THIS PROGRAM PROVE  DEFECTIVE, YOU ASSUME  THE COST  OF  ALL NECESSARY
     SERVICING, REPAIR OR CORRECTION.

     IN NO  EVENT  SHALL THE UNIVERSITY  OF TEXAS OR  ANY  OF ITS COMPONENT
     INSTITUTIONS INCLUDING M. D.   ANDERSON HOSPITAL BE LIABLE  TO YOU FOR
```

DAMAGES, INCLUDING ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA OR ITS ANALYSIS BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES) THE PROGRAM.

(Above NO WARRANTY modified from the GNU NO WARRANTY statement.)

B.3 Common Public License

Common Public License Version 0.5

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor

hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance

claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. EACH RECIPIENT IS SOLELY RESPONSIBLE FOR DETERMINING THE APPROPRIATENESS OF USING AND DISTRIBUTING THE PROGRAM AND ASSUMES ALL RISKS ASSOCIATED WITH ITS EXERCISE OF RIGHTS UNDER THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO THE RISKS AND COSTS OF PROGRAM ERRORS, COMPLIANCE WITH APPLICABLE LAWS, DAMAGE TO OR LOSS OF DATA, PROGRAMS OR EQUIPMENT, AND UNAVAILABILITY OR INTERRUPTION OF OPERATIONS.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as

reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

B.4 ZIB Academice License

ZIB ACADEMIC LICENSE

This license for ZIB software is designed to guarantee freedom to share and change software for academic use, but restricting commercial firms from exploiting your knowhow for their benefit. The precise terms and conditions for using, copying, distribution, and modification follow.

Terms and Conditions for Using, Copying, Distribution, and Modification

The "Program" below refers to source, object and executable code, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. Each licensee is addressed as "you".

1. This license applies to you only if you are a member of a noncommercial and academic institution, e.g., a university. The license expires as soon as you are no longer a member of this institution.
2. Every publication and presentation for which work based on the Program

```
   or its output has been used must contain an appropriate citation
   and aknowledgement of the author(s) of the Program.
3. You may copy and distribute the Program or work based on the Program
   in source, object, or executable form provided that you also meet
   all of the following conditions:

   a. You must cause any work that you distribute or publish, that in whole
      or in part contains or is derived from the Program or any
      part thereof, to be licensed as a whole at no charge under the terms of
      this License. You must accompany it with this unmodified license text.

      These requirements apply to the Program or work based on the Program
      as a whole. If identifiable sections of that work are
      not derived from the Program, and can be reasonably considered
      independent and separate works in themselves, this License
      does not apply to those sections when you distribute them as
      separate works. But when you distribute the same sections as
      part of a whole which is a work based on the Program, the distribution
      of the whole must be on the terms of this License,
      whose permissions for other licensees extend to the entire whole and,
      thus, to each and every part regardless of who wrote it.

   b. You must cause the modified files to carry prominent notices stating
      that you changed the files and the date of any change.
   c. You must keep track of access to the Program (e.g., similar to the
      registration procedure at ZIB).
   d. You must accompany it with the complete corresponding machine-readable
      source code.

      The source code for a work means the preferred form of the work for
      making modifications to it. For an executable work,
      complete source code means all the source code for all modules it
      contains, plus any associated interface definition files, plus
      the scripts used to control compilation and installation of the
      executable. However, as a special exception, the source code
      distributed need not include anything that is normally distributed
      (in either source or binary form) with the major components
      (compiler, kernel, and so on) of the operating system on which the
      executable runs, unless that component itself accompanies
      the executable.

4. You may not copy, modify, sublicense, or distribute the Program except as
   expressly provided under this License. Any attempt otherwise to
   copy, modify, sublicense, or distribute the Program is void and will
   automatically terminate your rights under this License. However, parties
   who have received copies or rights from you under this License will
   not have their licenses terminated so long as such parties remain in full
   compliance.
5. You are not required to accept this License, since you have not signed it.
   However, nothing else grants you permission to use, modify, or
   distribute the Program or its derivative works. These actions are
   prohibited by law if you do not accept this License. Therefore, by using,
   modifying or distributing the Program (or any work based on the Program),
   you indicate your acceptance of this License to do so and all its
   terms and conditions for copying, distributing or modifying the Program
   or works based on it.
6. Each time you redistribute the Program (or any work based on the Program),
   the recipient automatically receives a license from the original
   licensor to copy, distribute or modify the Program subject to these terms
```

```
   and conditions. You may not impose any further restrictions on the
   recipient's exercise of the rights granted herein. You are not responsible
   for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent
   infringement or for any other reason (not limited to patent issues),
   conditions are imposed on you (whether by court order, agreement, or
   otherwise) that contradict the conditions of this License, they do not
   excuse you from the conditions of this License.
8. If you wish to incorporate parts of the Program into other programs whose
   distribution conditions are different, write to ZIB to ask for permission.

NO WARRANTY

9. Because the program is licensed free of charge, there is no warranty for
   the program to the extent permitted by applicable law. The
   copyright holders provide the program "as is" without warranty of any
   kind, either expressed or implied, including, but not limited to, the
   implied warranties of merchantability and fitness for a particular purpose.
   The entire risk as to the quality and performance of the program is
   with you. Should the program prove defective, you assume the cost of all
   necessary servicing, repair, or correction.
10.In no event will any copyright holder, or any other party who may modify
   and/or redistribute the program as permitted above, be liable to
   you for damages, including any general, special, incidental or
   consequential damages arising out of the use or inability to use the
   program (including but not limited to loss of data or data being rendered
   inaccurate or losses sustained by you or third parties or a failure of the
   program to operate with any other programs), even if such holder or other
   party has been advised of the possibility of such damages.
```

B.5 Lucent Technologies License

```
/************************************************************
Copyright (C) 1992 Lucent Technologies
All Rights Reserved

Permission to use, copy, modify, and distribute this software and
its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all
copies and that both that the copyright notice and this
permission notice and warranty disclaimer appear in supporting
documentation, and that the name of Lucent or any of its entities
not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission.

LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS.
IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER
IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF
THIS SOFTWARE.
************************************************************
```

B.6 The GNU Lesser General Public License

```
                GNU LESSER GENERAL PUBLIC LICENSE
                     Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
     59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
```

author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the

user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any

attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License. However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not
signed it. However, nothing else grants you permission to modify or
distribute the Library or its derivative works. These actions are
prohibited by law if you do not accept this License. Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions. You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all. For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices. Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus

excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of

```
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

That's all there is to it!